

A SEMI-DECIDABLE PROCEDURE FOR SECRECY IN CRYPTOGRAPHIC PROTOCOLS

Jaouhar Fattahi¹ and Mohamed Mejri¹ and Hanane Houmani²

¹LSI Group, Laval University, Quebec, Canada

²University Hassan II, Morocco

ABSTRACT

In this paper, we present a new semi-decidable procedure to analyze cryptographic protocols for secrecy based on a new class of functions that we call: the Witness-Functions. A Witness-Function is a reliable function that guarantees the secrecy in any protocol proved increasing once analyzed by it. Hence, the problem of correctness becomes a problem of protocol growth. A Witness-Function operates on derivative messages in a role-based specification and introduces new derivation techniques. We give here the technical aspects of the Witness-Functions and we show how to use them in a semi-decidable procedure. Then, we analyze a variation of the Needham-Schroeder protocol and we show that a Witness-Function can also help to teach about flaws. Finally, we analyze the NSL protocol and we prove that it is correct with respect to secrecy.

KEYWORDS

Cryptographic Protocols, Decidability, Role-based specification, Secrecy

1 INTRODUCTION

In this paper, we present a new semi-decidable procedure for analyzing cryptographic protocols statically for the property of secrecy in a role-based specification. The main idea of this procedure is to prove the secrecy of a protocol by proving that it is increasing. Intuitively, an increasing protocol preserves secret. That means if the level of security of all atomic messages exchanged in the protocol does not decay between all receiving and sending steps in the protocol, the secret is preserved. For that, we need reliable metrics to estimate the level of security of atomic messages. This way of seeing secrecy in protocols has been adopted in some prior works. For instance, in [1], Steve Schneider suggested the rank-functions to analyze protocols in CSP [2, 3]. These functions were efficient for analyzing several protocols such the Needham-Schroeder protocol. However, using these functions dictates the protocol implementation in the CSP algebra. Besides, building these functions is not easy and their existence is not always possible [4]. In [5–8], Houmani et al. presented universal functions called interpretation functions to statically analyze a protocol for secrecy. An interpretation function needs to meet some conditions to be "good enough" to run an analysis. They were successful to analyze many protocols. However, we note that the conditions on these functions were very restrictive. That's why only two functions had been given: DEK and DEKAN. Naturally, less we have restrictions on functions, more we have chance to define many of them and therefore to prove the correctness of a larger range of protocols. In fact, one function may fail to prove the growth of a protocol but another may do so. In this regard, we think that the condition of full-invariance by substitution in Houmani's work is the most blocking one. This condition is though very important since it enables any decision made on messages of the generalized roles (messages with variables) to be propagated to valid traces (closed messages). Since the goal of our approach is to build as many functions as we can, we believe that if we liberate a function from this condition, we will be able to build several functions. However, liberating a function from a condition may oblige us to take extra precautions when using it. In this paper, we present the Witness-Functions as new metrics to analyze cryptographic protocols. A Witness-Function is tightly linked to an interpretation function but does not need the full-invariant by substitution property. In fact, a Witness-Function provides two attractive bounds that are independent of substitution. This fully replaces any need to this property. We also introduce the notion of derivative messages by using new derivation techniques. We exhibit the theorem of protocol analysis with the Witness-Functions. This theorem defines

a semi-decidable procedure for analyzing cryptographic protocols. Finally, we run an analysis on two protocols. First, we run an analysis on a variation of Needham-Schroeder protocol in which we show that a Witness-Function could even teach about flaws. Then, we run an analysis on NSL protocol where we prove that it is correct with respect to secrecy.

2 PRELIMINARY AND NOTATIONS

Here, we give some conventions and notations that we use in this paper.

+ We denote by $\mathcal{C} = \langle \mathcal{M}, \xi, \models, \mathcal{K}, \mathcal{L}^\sqsupset, \ulcorner, \urcorner \rangle$ the context of verification in which our analysis is run. It contains the parameters that affect the analysis of a protocol:

- \mathcal{M} : is a set of messages built from the signature $\langle \mathcal{N}, \Sigma \rangle$ where \mathcal{N} is a set of atomic names (nonces, keys, principals, etc.) and Σ is a set of functions (*enc*: encryption, *dec*: decryption, *pair*: concatenation (that we denote by "." here), etc.). i.e. $\mathcal{M} = T_{\langle \mathcal{N}, \Sigma \rangle}(\mathcal{X})$. We denote by Γ the set of substitutions from $\mathcal{X} \rightarrow \mathcal{M}$. We denote by \mathcal{A} all the atomic messages in \mathcal{M} , by $\mathcal{A}(m)$ the set of atomic messages (or atoms) in m and by \mathcal{I} the set of principals including the intruder I . We denote by k^{-1} the reverse form of a key k and we assume that $(k^{-1})^{-1} = k$.
- ξ : is the equational theory in which the algebraic properties of the functions in Σ are described by equations. e.g. $\text{dec}(\text{enc}(x, y), y^{-1}) = x$.
- $\models_{\mathcal{C}}$: is the inference system of the intruder under the equational theory. Let M be a set of messages and m a message. $M \models_{\mathcal{C}} m$ means that the intruder is able to infer m from M using her capacity. We extend this notation to traces as follows: $\rho \models_{\mathcal{C}} m$ means that the intruder can infer m from the messages exchanged in the trace ρ . We suppose that the intruder has the full control of the net as described by Dolev-Yao model in [9]. That is to say that she can intercept, delete, redirect and modify messages. She knows the public keys of all agents. She knows her private keys and the keys that she shares with other agents. She can encrypt or decrypt any message with known keys. Generically, the intruder has the following rules of building messages:

$$\begin{aligned}
 (\text{int}) : & \frac{\Box}{M \models_{\mathcal{C}} m} [m \in M \cup K(I)] \\
 (\text{op}) : & \frac{M \models_{\mathcal{C}} m_1, \dots, M \models_{\mathcal{C}} m_n}{M \models_{\mathcal{C}} f(m_1, \dots, m_n)} [f \in \Sigma] \\
 (\text{eq}) : & \frac{M \models_{\mathcal{C}} m', m' =_{\mathcal{C}} m}{M \models_{\mathcal{C}} m}, \text{ with } (m' =_{\mathcal{C}} m) \equiv (m' =_{\xi(\mathcal{C})} m)
 \end{aligned}$$

Example 2.1.

The intruder capacity can be described by the following rules:

$$\begin{aligned}
 (\text{int}) : & \frac{\Box}{M \models_{\mathcal{C}} m} [m \in M \cup K(I)] \\
 (\text{concat}) : & \frac{M \models_{\mathcal{C}} m_1, M \models_{\mathcal{C}} m_2}{M \models_{\mathcal{C}} m_1.m_2} \\
 (\text{deconcat}) : & \frac{M \models_{\mathcal{C}} m_1.m_2}{M \models_{\mathcal{C}} m_i} [i \in \{1, 2\}] \\
 (\text{dec}) : & \frac{M \models_{\mathcal{C}} k, M \models_{\mathcal{C}} m_k}{M \models_{\mathcal{C}} m} \\
 (\text{enc}) : & \frac{M \models_{\mathcal{C}} k, M \models_{\mathcal{C}} m}{M \models_{\mathcal{C}} \{m\}_k}
 \end{aligned}$$

In this example, from a set of messages, an intruder can infer any message in this set. She can encrypt any message when she holds the encryption key. She can decrypt any message when she holds the decryption key and concatenate any two messages and deconcatenate them.

- \mathcal{K} : is a function from \mathcal{I} to \mathcal{M} , that returns to any agent a set of atomic messages describing her initial knowledge. We denote by $K_{\mathcal{C}}(I)$ the initial knowledge of the intruder, or simply $K(I)$ where the context is obvious.
- \mathcal{L}^\sqsupset : is the lattice of security ($\mathcal{L}, \sqsupseteq, \sqsubset, \sqcap, \sqcup, \top$) used to assign security levels to messages. An example of a lattice is $(2^{\mathcal{I}}, \subseteq, \cap, \cup, \mathcal{I}, \emptyset)$ that will be used to attribute to an atomic message α the set of agents that are authorized to know it.

- $\ulcorner \cdot \urcorner$: is a partial function that attributes a value of security (or type) to a message in \mathcal{M} . Let M be a set of messages and m a message. We write $\ulcorner M \urcorner \sqsupseteq \ulcorner m \urcorner$ if $\exists m' \in M. \ulcorner m' \urcorner \sqsupseteq \ulcorner m \urcorner$
- + Our analysis is performed in a role-based specification. A role-based specification is a set of generalized roles. A generalized role is an abstraction of the protocol where the emphasis is put on a specific agent and where all the unknown messages, and on which the agent cannot carry out any verification, are substituted by variables. An exponent i (the session identifier) is added to a fresh message to say that these components change values from one run to another. A generalized role interprets how a particular agent understands the exchanged messages. We extract it from a protocol as follows:
 - we extract the roles from the protocol.
 - we substitute the unknown messages by fresh variables for each role.

The roles are extracted as follows:

- For each agent, we extract from the protocol all the steps in which this principal participates. Then, we add to this abstraction a session identifier i in the steps identifiers and in the fresh values.
- We introduce an intruder I to express the fact that the received messages and the sent messages are probably sent or received by the intruder.
- Finally, we extract all prefixes from those roles where a prefix ends by a sending step.

From the roles, we generate the generalized roles. In a generalized role, unknown messages are substituted by variables to express that the agent cannot be sure about its integrity or its origin. The role-based specification expresses the notion of valid traces of a protocol. More details about the role-based specification could be found in [10–13].

Example 2.2.

Let us consider the Needham-Schroeder protocol given in Table 1.

The generalized roles of the agent A are:

Table 1: The Needham-Schroeder Protocol

$p_1 = \begin{aligned} &\langle 1, A \rightarrow B : \{A.N_a\}_{k_b} \rangle. \\ &\langle 2, B \rightarrow A : \{N_a.N_b.B\}_{k_a} \rangle. \\ &\langle 3, A \rightarrow B : \{N_b\}_{k_b} \rangle. \end{aligned}$
--

$$\begin{aligned} \mathcal{A}_G^1 &= \langle i.1, A \rightarrow I(B) : \{A.N_a^i\}_{k_b} \rangle \\ \mathcal{A}_G^2 &= \langle i.1, A \rightarrow I(B) : \{A.N_a^i\}_{k_b} \rangle. \\ &\quad \langle i.2, I(B) \rightarrow A : \{N_a^i.X.B\}_{k_a} \rangle. \\ &\quad \langle i.3, A \rightarrow I(B) : \{X\}_{k_b} \rangle \end{aligned}$$

The generalized roles of the agent B are:

$$\begin{aligned} \mathcal{B}_G^1 &= \langle i.1, I(A) \rightarrow B : \{A.Y\}_{k_b} \rangle. \\ &\quad \langle i.2, B \rightarrow I(A) : \{Y.N_b^i.B\}_{k_a} \rangle \\ \mathcal{B}_G^2 &= \langle i.1, I(A) \rightarrow B : \{A.Y\}_{k_b} \rangle. \\ &\quad \langle i.2, B \rightarrow I(A) : \{Y.N_b^i.B\}_{k_a} \rangle. \\ &\quad \langle i.3, I(A) \rightarrow B : \{N_b^i\}_{k_b} \rangle \end{aligned}$$

The role-based specification of the protocol in Table 1 is $\mathcal{R}_G(p_1) = \{\mathcal{A}_G^1, \mathcal{A}_G^2, \mathcal{B}_G^1, \mathcal{B}_G^2\}$.

Example 2.3.

Let us consider the NSL protocol given in Table 2.

The generalized roles of the agent A are:

Table 2: The NSL Protocol

$$p_2 = \begin{array}{l} \langle 1, A \rightarrow B : \{N_a.A\}_{k_b} \rangle. \\ \langle 2, B \rightarrow A : \{B.N_a\}_{k_a} \cdot \{B.N_b\}_{k_a} \rangle. \\ \langle 3, A \rightarrow B : A.B.\{N_b\}_{k_b} \rangle \end{array}$$

$$\mathcal{A}'_G^1 = \langle i.1, A \rightarrow I(B) : \{N_a^i.A\}_{k_b} \rangle$$

$$\mathcal{A}'_G^2 = \begin{array}{l} \langle i.1, A \rightarrow I(B) : \{N_a^i.A\}_{k_b} \rangle. \\ \langle i.2, I(B) \rightarrow A : \{B.N_a^i\}_{k_a} \cdot \{B.X\}_{k_a} \rangle. \\ \langle i.3, A \rightarrow I(B) : A.B.\{X\}_{k_b} \rangle \end{array}$$

The generalized roles of the agent B are:

$$\mathcal{B}'_G^1 = \begin{array}{l} \langle i.1, I(A) \rightarrow B : \{Y.A\}_{k_b} \rangle. \\ \langle i.2, B \rightarrow I(A) : \{B.Y\}_{k_a} \cdot \{B.N_b^i\}_{k_a} \rangle \end{array}$$

$$\mathcal{B}'_G^2 = \begin{array}{l} \langle i.1, I(A) \rightarrow B : \{Y.A\}_{k_b} \rangle. \\ \langle i.2, B \rightarrow I(A) : \{B.Y\}_{k_a} \cdot \{B.N_b^i\}_{k_a} \rangle. \\ \langle i.3, I(A) \rightarrow B : A.B.\{N_b^i\}_{k_b} \rangle \end{array}$$

The role-based specification of the protocol in Table 2 is $\mathcal{R}_G(p_2) = \{\mathcal{A}'_G^1, \mathcal{A}'_G^2, \mathcal{B}'_G^1, \mathcal{B}'_G^2\}$.

- + A valid trace is an interleaving of substituted generalized roles where each message sent by the intruder can be generated by her using her capacity and by the received messages. We denote by $\llbracket p \rrbracket$ the set of valid traces generated by p .
- + We denote by \mathcal{M}_p^G the set of messages (with variables) in $R_G(p)$, by \mathcal{M}_p the set of closed messages generated by substitution in \mathcal{M}_p^G . We denote by R^+ (respectively R^-) the set of sent messages (respectively received messages) by a honest agent in the role R . Conventionally, we devote the uppercase symbols for sets or sequences of elements and the lowercase for single elements. For example, M denotes a set of messages, m a single message, R a role composed of a sequence of steps, r a step and $R.r$ the role ending by the step r .
- + In our analysis, no restriction on the size of messages or the number of sessions in the protocols is made.

3 INCREASING PROTOCOLS DO NOT REVEAL SECRETS

To analyze a protocol, we need functions to estimate the security level of every atomic message. In this section, we give sufficient conditions on a function F to guarantee that it is enough good (or reliable) to run an analysis and we show that an increasing protocol is correct with respect to the secrecy property when analyzed with such functions.

3.1 \mathcal{C} -reliable functions

A function F is said to be well-formed when it returns the lowest value in the lattice, denoted by \perp , for an atomic message α that appears in clear. It returns for it in the union of two sets, the minimum " \sqcap " of the two values calculated in each set separately. It returns the uppermost value, denoted by " \sqcup ", if it does not appear in this set. These facts are expressed by the definition 3.1.

Definition 3.1. (Well-formed function)

Let F be a function and \mathcal{C} a context of verification.

F is well-formed in \mathcal{C} if:

$$\forall M, M_1, M_2 \subseteq \mathcal{M}, \forall \alpha \in \mathcal{A}(\mathcal{M}):$$

$$\begin{cases} F(\alpha, \{\alpha\}) & = \perp \\ F(\alpha, M_1 \cup M_2) & = F(\alpha, M_1) \sqcap F(\alpha, M_2) \\ F(\alpha, M) & = \sqcup, \text{ if } \alpha \notin \mathcal{A}(M) \end{cases}$$

A function F is said to be full-invariant-by-intruder if when it attributes a security level to a message α in a set of messages M , the intruder can never produce another message m that decrease this level (i.e. $F(\alpha, m) \supseteq F(\alpha, M)$) using her capacity in the context of verification, except when α is intended to be known by the intruder (i.e. $\ulcorner K(I) \urcorner \supseteq \ulcorner \alpha \urcorner$). This fact is expressed by the definition 3.2.

Definition 3.2. (Full-invariant-by-intruder function)

Let F be a function and \mathcal{C} a context of verification.

F is full-invariant-by-intruder in \mathcal{C} if:

$$\forall M \subseteq \mathcal{M}, m \in \mathcal{M}. M \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m). (F(\alpha, m) \supseteq F(\alpha, M)) \vee (\ulcorner K(I) \urcorner \supseteq \ulcorner \alpha \urcorner)$$

A function F is said to be reliable if it is well-formed and full-invariant-by-intruder. This fact is expressed by the definition 3.3.

Definition 3.3. (Reliable function)

Let F be a function and \mathcal{C} a context of verification.

F is \mathcal{C} -reliable if F is well-formed and F is full-invariant-by-intruder in \mathcal{C} .

A protocol p is said to be F -increasing when every principal generates increasingly valid traces (substituted generalized roles) that never decrease the security levels of received components. The estimation of the value of security of every atom is performed by F . This fact is expressed by the definition 3.4.

Definition 3.4. (F -increasing protocol)

Let F be a function, \mathcal{C} a context of verification and p a protocol.

p is F -increasing in \mathcal{C} if:

$\forall R.r \in R_G(p), \forall \sigma \in \Gamma : \mathcal{X} \rightarrow \mathcal{M}_p$ we have:

$$\forall \alpha \in \mathcal{A}(\mathcal{M}_p). F(\alpha, r^+ \sigma) \supseteq \ulcorner \alpha \urcorner \sqcap F(\alpha, R^- \sigma)$$

A secret disclosure consists in manipulating a valid trace of the protocol (denoted by $\llbracket p \rrbracket$) by the intruder using her knowledge $K(I)$ in a context of verification \mathcal{C} , to deduce a secret α that she is not intended to know (expressed by: $\ulcorner K(I) \urcorner \not\supseteq \ulcorner \alpha \urcorner$). This fact is expressed by the definition 3.5.

Definition 3.5. (Secret disclosure)

Let p be a protocol and \mathcal{C} a context of verification.

We say that p discloses a secret $\alpha \in \mathcal{A}(\mathcal{M})$ in \mathcal{C} if:

$$\exists \rho \in \llbracket p \rrbracket. (\rho \models_{\mathcal{C}} \alpha) \wedge (\ulcorner K(I) \urcorner \not\supseteq \ulcorner \alpha \urcorner)$$

Lemma 3.6.

Let F be a \mathcal{C} -reliable function and p an F -increasing protocol.

We have:

$$\forall m \in \mathcal{M}. \llbracket p \rrbracket \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m). (F(\alpha, m) \supseteq \ulcorner \alpha \urcorner) \vee (\ulcorner K(I) \urcorner \supseteq \ulcorner \alpha \urcorner)$$

See the proof 4 in [14]

The lemma 3.6 says that for any atom α in a message produced by an increasing protocol, its security level returned by a reliable function is kept greater or equal than its initial value in the context, if the intruder is not initially allowed to know it. Hence, initially the atom has a certain level of security. This value cannot be decreased by the intruder using her knowledge and the received messages since it is full-invariant-by-intruder. In every new step of a valid trace, involved messages are better protected since the protocol is increasing. The proof is then run by induction on the size of the trace using the reliability properties of the function in every step of the induction.

Theorem 3.7. (Theorem of Correctness of Increasing Protocols)

Let F be a \mathcal{C} -reliable function and p a F -increasing protocol.

p is \mathcal{C} -correct with respect to the secrecy property.

Proof.

Let's suppose that p discloses an atomic secret α .

From the definition 3.5 we have:

$$\exists \rho \in \llbracket p \rrbracket. (\rho \models_{\mathcal{C}} \alpha) \wedge (\ulcorner K(I) \urcorner \not\sqsubseteq \ulcorner \alpha \urcorner) \quad (1)$$

Since F is a \mathcal{C} -reliable function and p an F -increasing protocol, we have from the lemma 3.6:

$$(F(\alpha, \alpha) \sqsubseteq \ulcorner \alpha \urcorner) \vee (\ulcorner K(I) \urcorner \sqsubseteq \ulcorner \alpha \urcorner) \quad (2)$$

From 1 and 2, we have:

$$F(\alpha, \alpha) \sqsubseteq \ulcorner \alpha \urcorner \quad (3)$$

Since F is well-formed in \mathcal{C} , then:

$$F(\alpha, \alpha) = \perp \quad (4)$$

From 3 and 4 we have:

$$\perp = \ulcorner \alpha \urcorner \quad (5)$$

5 is impossible because it is contradictory with: $\ulcorner K(I) \urcorner \not\sqsubseteq \ulcorner \alpha \urcorner$ in 1.

Then p is \mathcal{C} -correct with respect to the secrecy property.

Theorem 3.1 asserts that an increasing protocol is correct with respect to the secrecy property when analyzed with a reliable function. It is worth saying that compared to the sufficient conditions stated in [8], we have one less. Thus, in [8], Houmani demanded from the function an additional condition: the full-invariance by substitution. That's to say, the interpretation function has also to resist to the problem of substitution of variables. Here, we liberate our functions from this blocking condition in order to be able to build more of them. We rehouse this condition in our new definition of an increasing protocol which is required now to be increasing on valid traces (closed messages) rather than messages of the generalized roles (message with variables). Therefore, the problem of substitutions is transferred to the protocol and becomes less difficult to handle.

4 CONSTRUCTION OF RELIABLE INTERPRETATION FUNCTIONS

As seen in the previous section, to analyze a protocol we need reliable interpretation functions to estimate the level of security of any atom in a message. In this section, we exhibit a constructive way to build these functions. We first exhibit the way to build a generic class of reliable selections inside the protection of the most external key (or simply the external key). Then we propose specialized selections of this class. Finally we give the way to build reliable selection-based interpretation functions. Similar techniques based on selections were proposed in previous works, especially in [5, 7, 8] to build universal functions based on the selection of the direct key of encryption and in [15] to check correspondences in protocols. But first of all, we present the notion of well-protected messages that have valuable properties that we will use in the definition of reliable selections.

4.1 Protocol analysis in Well-Protected Messages

4.1.1 Well-Protected Messages

Briefly, a well-protected message is a message such that every non public atom α in it is encrypted by at least one key k such that $\ulcorner k^{-1} \urcorner \sqsubseteq \ulcorner \alpha \urcorner$, after elimination of unnecessary keys (e.g. $e(k, d(k^{-1}, m)) \rightarrow m$). The main advantage of an analysis performed over a set of well-protected messages is that the intruder cannot deduce any secret when she uses only her knowledge in the context of verification (without using the protocol rules).

Lemma 4.1.

Let M be a set of well-protected messages in \mathcal{M} . We have:

$$M \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m). (\alpha \text{ is well-protected in } m) \vee (\ulcorner K(I) \urcorner \sqsubseteq \ulcorner \alpha \urcorner)$$

Lemma 4.1 says that from a set of well-protected messages, all atomic messages beyond the knowledge of the intruder (i.e. $\lceil K(I) \rceil \not\supseteq \lceil \alpha \rceil$) remain well-protected in any message that the intruder could infer. Indeed, since each atom that does not appear in clear (non-public) in this set is encrypted by at least one key k such that $\lceil k^{-1} \rceil \supseteq \lceil \alpha \rceil$, then the intruder has to retrieve the key k^{-1} before she sees α not well-protected in any message (clear). But, the key k^{-1} is in its turn encrypted by at least one key k' such that $\lceil k'^{-1} \rceil \supseteq \lceil k^{-1} \rceil$. The proof is then conducted by induction on the encryption keys.

Lemma 4.2. (Lemma of non-disclosure of atomic secrets in well-protected messages)

Let M be a set of well-protected messages in \mathcal{M} and α an atomic message in M .

We have:

$$M \models_{\mathcal{C}} \alpha \Rightarrow \lceil K(I) \rceil \supseteq \lceil \alpha \rceil$$

Proof.

From Lemma 4.1, we have $M \models_{\mathcal{C}} \alpha$ then:

$$(\alpha \text{ is well-protected in } \alpha) \vee (\lceil K(I) \rceil \supseteq \lceil \alpha \rceil) \quad (6)$$

But α is not well-protected in α , then we have:

$$\lceil K(I) \rceil \supseteq \lceil \alpha \rceil \quad (7)$$

4.2 Discussion and Assumption

Lemma 4.2 expresses an important result. It states that from a set of well-protected messages the intruder cannot deduce any secret that she is not supposed to know when she uses only her knowledge in the context of verification (without using the protocol rules). It is worth saying that verifying whether a protocol operates over a space of well-protected messages or not is an easy task and most of real protocols respect this condition.

4.3 Building reliable selections

Now, we will focus on building selections such that when they are composed to suitable homomorphisms, provide reliable interpretation functions. The definition 4.3 introduces the notion of a well-formed selection and the definition 4.4 introduces the notion of a full-invariant-by-intruder selection.

Definition 4.3. (Well-formed selection)

Let $M, M_1, M_2 \subseteq \mathcal{M}$ such that M, M_1 and M_2 are well-protected.

Let $S : \mathcal{A} \times \mathcal{M} \mapsto 2^{\mathcal{A}}$ be a selection.

We say that S is well-formed in \mathcal{C} if:

$$\begin{cases} S(\alpha, \{\alpha\}) &= \mathcal{A}, \\ S(\alpha, M_1 \cup M_2) &= S(\alpha, M_1) \cup S(\alpha, M_2), \\ S(\alpha, M) &= \emptyset, \text{ if } \alpha \notin \mathcal{A}(M) \end{cases}$$

For an atom α in a set of messages M , a well-formed selection returns all the atoms in \mathcal{M} if $M = \{\alpha\}$. It returns for it in the union of two sets of messages, the union of the two selections performed in each set separately. It returns the empty set if the atom does not appear in M .

Definition 4.4. (Full-invariant-by-intruder selection)

Let $M \subseteq \mathcal{M}$ such that M is well-protected.

Let $S : \mathcal{A} \times \mathcal{M} \mapsto 2^{\mathcal{A}}$ be a selection.

We say that S is full-invariant-by-intruder in \mathcal{C} if:

$\forall M \subseteq \mathcal{M}, m \in M$, we have:

$$M \models_{\mathcal{C}} m \Rightarrow \forall \alpha \in \mathcal{A}(m). (S(\alpha, m) \subseteq S(\alpha, M)) \vee (\lceil K(I) \rceil \supseteq \lceil \alpha \rceil)$$

The aim of a full-invariant-by-intruder selection is to provide a full-invariant-by-intruder function when composed to an adequate homomorphism that transforms its returned atoms into security levels. Since a

full-invariant-by-intruder function is requested to resist to any attempt of the intruder to generate a message m from any set of messages M in which the level of security of an atom, that she is not allowed to know, decreases compared to its value in M , a full-invariant-by-intruder selection is requested to resist to any attempt of the intruder to generate a message m from any set of messages M in which the selection associated to an atom, that she is not allowed to know, could be enlarged compared to the selection associated to this atom in M . This fact is described by the definition 4.4.

Definition 4.5. (Reliable selection)

Let $S : \mathcal{A} \times \mathcal{M} \mapsto 2^{\mathcal{A}}$ be a selection and \mathcal{C} be a context of verification.

S is \mathcal{C} -reliable if S is well-formed and S is full-invariant-by-intruder in \mathcal{C} .

4.3.1 Reliable selections inside the protection of an external key

Here, we define a generic class of selections that we denote by S_{Gen}^{EK} and we prove that any instance of it is reliable under some condition.

Definition 4.6. (S_{Gen}^{EK} : selection inside the protection of an external key)

We denote by S_{Gen}^{EK} the class of all selections S that meet the following conditions:

- $S(\alpha, \alpha) = \mathcal{A}$; (8)

- $S(\alpha, m) = \emptyset$, if $\alpha \notin \mathcal{A}(m)$; (9)

- $\forall \alpha \in \mathcal{A}(m)$, where $m = f_k(m_1, \dots, m_n)$: (10)

$$S(\alpha, m) \subseteq \left(\bigcup_{1 \leq i \leq n} \mathcal{A}(m_i) \cup \{k^{-1}\} \setminus \{\alpha\} \right) \text{ if } f_k \in \mathcal{E}_{\mathcal{C}} \text{ and } \ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner \text{ and } m = m_{\Downarrow}$$

- $\forall \alpha \in \mathcal{A}(m)$, where $m = f(m_1, \dots, m_n)$: (11)

$$S(\alpha, m) = \begin{cases} \bigcup_{1 \leq i \leq n} S(\alpha, m_i) & \text{if } f_k \in \mathcal{E}_{\mathcal{C}} \text{ and } \ulcorner k^{-1} \urcorner \not\sqsupseteq \ulcorner \alpha \urcorner \text{ and } m = m_{\Downarrow} \text{ (a)} \\ \bigcup_{1 \leq i \leq n} S(\alpha, m_i) & \text{if } f \in \overline{\mathcal{E}}_{\mathcal{C}} \text{ and } m = m_{\Downarrow} \text{ (b)} \\ S(\alpha, m_{\Downarrow}) & \text{if } m \neq m_{\Downarrow} \text{ (c)} \end{cases}$$

- $S(\alpha, \{m\} \cup M) = S(\alpha, m) \cup S(\alpha, M)$ (12)

For an atom α in an encrypted message $m = f_k(m_1, \dots, m_n)$, a selection S as defined above returns a subset (see " \subseteq " in equation 10) among atoms that are neighbors of α in m inside the protection of the most external protective key k including its reverse form k^{-1} . The atom α itself is not selected. This set of candidate atoms is denoted by $\bigcup_{1 \leq i \leq n} \mathcal{A}(m_i) \cup \{k^{-1}\} \setminus \{\alpha\}$ in the equation 10. The most external protective key (or simply the external key) is the most external one that satisfies $\ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner$. A such key must exist when the set of messages generated by the protocol is well-protected, which is one of our assumptions above. By neighbor of α in m , we mean any atom that travels with it inside the protection of the external key.

S_{Gen}^{EK} defines a generic class of selections since it does not identify what atoms to select precisely inside the protection of the external key. It identifies only the atoms that are candidates for selection and among them we are allowed to return any subset.

Proposition 4.7.

Let $S \in S_{Gen}^{EK}$ and \mathcal{C} be a context of verification.

Let's have a rewriting system \rightarrow_{ξ} such that $\forall m \in \mathcal{M}, \forall \alpha \in \mathcal{A}(m) \wedge \alpha \notin \text{Clear}(m)$, we have:

$$\forall l \rightarrow r \in \rightarrow_{\xi}, S(\alpha, r) \subseteq S(\alpha, l) \quad (13)$$

We have:

S is \mathcal{C} -reliable.

Remark. The condition on the rewriting system \rightarrow_ξ given by the equation 13 in the definition 4.7 is introduced to make sure that the selection in the normal form is the smallest among all forms of a given message. This prevents the selection S to select atoms that are inserted maliciously by the intruder by manipulating the equational theory. Hence, we are sure that all selected atoms by S are honest. For example, let $m = \{\alpha.D\}_{k_{ab}}$ be a message in a homomorphic cryptography (i.e. $\{\alpha.D\}_{k_{ab}} = \{\alpha\}_{k_{ab}} \cdot \{D\}_{k_{ab}}$). In the form $\{\alpha.D\}_{k_{ab}}$, the selection $S(\alpha, \{\alpha.D\}_{k_{ab}})$ may select D since it is a neighbor of α inside the protection of k_{ab} , but in the form $\{\alpha\}_{k_{ab}} \cdot \{D\}_{k_{ab}}$ the selection $S(\alpha, \{\alpha\}_{k_{ab}} \cdot \{D\}_{k_{ab}})$ may not because it is not a neighbor of α . Then, we have to make sure that the rewriting system \rightarrow_ξ we are using is oriented in such way that it chooses the form $\{\alpha\}_{k_{ab}} \cdot \{D\}_{k_{ab}}$ rather than the form $\{\alpha.D\}_{k_{ab}}$ because there is no certitude that D had not been inserted maliciously by the intruder using the homomorphic property in the equational theory. We assume that the rewriting system we are using meets this condition.

As for Proposition 4.7, it is easy to check that by construction a selection S , that is instance of S_{Gen}^{EK} , is well-formed. The proof of full-invariance-by-intruder is carried out by induction on the tree of construction of a message. The principal idea of the proof is that the selection related to an atom α in a message m takes place inside the encryption by the most external protective key (such that: $\ulcorner k^{-1} \urcorner \sqsupseteq \ulcorner \alpha \urcorner$). Thus, an intruder cannot modify this selection when she does not have the key k^{-1} (i.e. $\ulcorner K(I) \urcorner \not\sqsupseteq \ulcorner k^{-1} \urcorner$). Besides, according to Lemma 4.2, in a set of well-protected messages the intruder can never infer this key since it is atomic. So, this selection can only be modified by people who are initially authorized to know α (i.e. $\ulcorner K \urcorner \sqsupseteq \ulcorner k^{-1} \urcorner$ and then $\ulcorner K \urcorner \sqsupseteq \ulcorner \alpha \urcorner$). In addition, the intruder cannot neither use the equational theory to alter this selection thanks to the condition made on the rewriting system in the remark 4.3.1. Therefore any set of candidate atoms returned by S cannot be altered (enlarged) by the intruder in any message m that she can infer, as required by a full-invariant-by-intruder selection.

Example 4.8.

Let α be an atomic message and m a message such that: $\ulcorner \alpha \urcorner = \{A, B\}$ and $m = \{A.C.\alpha.D\}_{k_{ab}}$. Let S_1, S_2 and S_3 be three selections such that: $S_1(\alpha, m) = \{k_{ab}^{-1}\}$, $S_2(\alpha, m) = \{A, C, k_{ab}^{-1}\}$ and $S_3(\alpha, m) = \{A, C, D, k_{ab}^{-1}\}$. These three selections are \mathcal{C} -reliable.

4.4 Instantiation of reliable selections from the class S_{Gen}^{EK}

Now that we defined a generic class of reliable selections S_{Gen}^{EK} , we will instantiate some concrete selections from it, that are naturally reliable. Instantiating S_{Gen}^{EK} consists in defining selections that return precise sets of atoms among the candidates allowed by S_{Gen}^{EK} .

4.4.1 The selection S_{MAX}^{EK}

The selection S_{MAX}^{EK} is the instance of the class S_{Gen}^{EK} that returns for an atom in a message m all its neighbors, that are principal identities, inside the protection of the external protective key k in addition to its reverse key k^{-1} . (MAX means: the MAXimum of principal identities)

4.4.2 The selection S_{EK}^{EK}

The selection S_{EK}^{EK} is the instance of the class S_{Gen}^{EK} that returns for an atom in a message m only the reverse key of the external protective key. (EK means: External Key)

4.4.3 The selection S_N^{EK}

The selection S_N^{EK} is the instance of the class S_{Gen}^{EK} that returns only its neighbors, that are principal identities, inside the protection of the external protective key. (N means: Neighbors)

Example 4.9.

Let α be an atom and m a message such that: $\ulcorner \alpha \urcorner = \{A, C\}$ and $m = \{\{\{\alpha.E\}_{k_{ab}}.F\}_{k_{ac}}.D\}_{k_{ad}}$
 $S_{MAX}^{EK}(\alpha, m) = \{E, F, k_{ac}^{-1}\}$; $S_{EK}^{EK}(\alpha, m) = \{k_{ac}^{-1}\}$; $S_N^{EK}(\alpha, m) = \{E, F\}$

4.5 Specialized \mathcal{C} -reliable selection-based interpretation functions

Now, we define specific functions that are a composition of an appropriate homomorphism and instances of the class of selections S_{Gen}^{EK} . This homomorphism exports the properties of reliability from a selection to a function and transforms selected atoms to security levels. The following proposition states that any function that is a composition of the homomorphism defined below and the selections S_{Gen}^{EK} is reliable.

Proposition 4.10.

Let ψ be a homomorphism defined as follows:

$$\begin{aligned} \psi &: (2^{\mathcal{A}})^{\subseteq} \mapsto \mathcal{L}^{\supseteq} \\ M &\mapsto \begin{cases} \top & \text{if } M = \emptyset \\ \bigcap_{\alpha \in M} \psi(\alpha) & \text{if not.} \end{cases} \end{aligned}$$

$$\text{such that: } \psi(\alpha) = \begin{cases} \{\alpha\} & \text{if } \alpha \in \mathcal{I} \text{ (Principal Identities)} \\ \ulcorner \alpha \urcorner & \text{if not.} \end{cases}$$

We have: $F_{MAX}^{EK} = \psi \circ S_{MAX}^{EK}$, $F_{EK}^{EK} = \psi \circ S_{EK}^{EK}$ and $F_N^{EK} = \psi \circ S_N^{EK}$ are \mathcal{C} -reliable.

The homomorphism ψ in Proposition 4.10 assigns for a principal in a selection, its identity. It assigns for a key its level of security in the context of verification. This homomorphism ensures the mapping from the operator " \subseteq " to the operator " \supseteq " in the lattice which offers to an interpretation function to inherit the full-invariance-by-intruder from its associated selection. In addition, it ensures the mapping from the operator " \cup " to the operator " \cap " in the lattice, which offers to an interpretation function to be well-formed if its associated selection is well-formed. Generally, every function $\psi \circ S$ remains reliable for any selection S in S_{Gen}^{EK} .

Example 4.11.

Let α be an atom, m a message and k_{ab} a key such that:

$$\begin{aligned} \ulcorner \alpha \urcorner &= \{A, B, S\}; m = \{A, C, \alpha, D\}_{k_{ab}}; \ulcorner k_{ab}^{-1} \urcorner = \{A, B, S\}; \\ S_{EK}^{EK}(\alpha, m) &= \{k_{ab}^{-1}\}; S_N^{EK}(\alpha, m) = \{A, C, D\}; S_{MAX}^{EK}(\alpha, m) = \{A, C, D, k_{ab}^{-1}\}; \\ F_{EK}^{EK}(\alpha, m) &= \psi \circ S_{EK}^{EK}(\alpha, m) = \ulcorner k_{ab}^{-1} \urcorner = \{A, B, S\}; F_N^{EK}(\alpha, m) = \psi \circ S_N^{EK}(\alpha, m) = \{A, C, D\}; \\ F_{MAX}^{EK}(\alpha, m) &= \psi \circ S_{MAX}^{EK}(\alpha, m) = \{A, C, D\} \cup \ulcorner k_{ab}^{-1} \urcorner = \{A, C, D\} \cup \{A, B, S\} = \{A, C, D, B, S\}. \end{aligned}$$

5 INSUFFICIENCY OF RELIABLE FUNCTION TO ANALYZE GENERALIZED ROLES

So far, we presented a class of selection-based functions that have the required properties to analyze protocols statically. Unluckily, they operate on valid traces that contain closed messages only. Nevertheless, a static analysis must be led over the finite set of messages of the generalized roles of the protocol because the set of valid traces is infinite. The problem is that the finite set of the generalized roles contains variables and the functions we defined are not "enough prepared" to analyze such messages because they are not supposed to be full-invariant by substitution [16–18]. The full-invariance by substitution is the property that allows us to perform an analysis over messages with variables and to export the conclusion made-on to closed messages. In the following section, we deal with the substitution question. We introduce the concept of derivative messages to reduce the impact of variables and we build the Witness-Functions that operate on these derivative messages rather than messages themselves. As we will see, the Witness-Functions provide two interesting bounds that are independent of all substitutions. This fully replaces the property of full-invariance by substitution. Finally, we define a criterion of protocol correctness based on these two bounds.

5.1 Derivative message

Let $m, m_1, m_2 \in \mathcal{M}$; $\mathcal{X}_m = Var(m)$; $S_1, S_2 \subseteq 2^{\mathcal{X}_m}$; $\alpha \in \mathcal{A}(m)$; $X, Y \in \mathcal{X}_m$ and ϵ be the empty message.

Definition 5.1. (Derivation)

We define the derivative message as follows:

$$\begin{aligned}
\partial_X \epsilon &= \epsilon \\
\partial_X \alpha &= \alpha \\
\partial_X X &= \epsilon \\
\partial_X Y &= Y, X \neq Y \\
\partial_X f(m) &= f(\partial_X m), f \in \mathcal{E}_C \cup \bar{\mathcal{E}}_C \\
\partial_{\{X\}} m &= \partial_X m \\
\partial(\overline{X}) m &= \partial_{\{X_m \setminus X\}} m \\
\partial_{S_1 \cup S_2} m &= \partial_{S_2 \cup S_1} m = \partial_{S_1} \partial_{S_2} m = \partial_{S_2} \partial_{S_1} m
\end{aligned}$$

To be simple, we denote by ∂m the expression $\partial_{X_m} m$. The operation of derivation introduced by the definition 5.1 (denoted by ∂) eliminates variables in a message. $\partial_X m$ consists in eliminating the variable X in m . $\partial(\overline{X}) m$ consists in eliminating all variables, except X , in m . Hence, X when overlined is considered as a constant in m . ∂m consists in eliminating all the variables in m .

Definition 5.2.

Let $m \in \mathcal{M}_p^G$, $X \in \mathcal{X}_m$ and $m\sigma$ be a closed message.

For all $\alpha \in \mathcal{A}(m\sigma)$, $\sigma \in \Gamma$, we denote by:

$$F(\alpha, \partial[\overline{\alpha}]m\sigma) = \begin{cases} F(\alpha, \partial m) & \text{if } \alpha \in \mathcal{A}(\partial m), \\ F(X, \partial[\overline{X}]m) & \text{if } \alpha = X\sigma \wedge \alpha \notin \mathcal{A}(\partial m). \end{cases}$$

A message m in a generalized role is composed of two parts: a static part and a dynamic part. The dynamic part is described by variables. For an atom α in the static part (i.e. ∂m), $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ removes the variables in m and gives it the value $F(\alpha, \partial m)$. For anything that is not an atom of the static part -that comes by substitution of some variable X in m - $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ considers it as the variable itself, treated as a constant and as a block, and gives it all the time the same value: $F(X, \partial[\overline{X}]m)$. For any F such that its associated selection is an instance of the class S_{Gen}^{EK} , $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ depends only on the static part of m since α is not selected. The function in the definition 5.2 presents the three following major facts:

1. An atom of the static part of a message with variables, when analyzed with a such function, is considered as an atom in a message with no variables (a closed message);
2. A variable, when analyzed by such function, is considered as any component that substitutes it (that is not in the static part of the message) with no respect to other variables, if any;
3. For any F such that its associated selection is an instance of the class S_{Gen}^{EK} , $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ depends only on the static part of m since α is not selected.

One could suggest that we attribute to an atom α in a closed message $m\sigma$ the value returned by the function $F(\alpha, \partial[\overline{\alpha}]m\sigma)$ given in the definition 5.2 and hence we neutralize the variable effects. Unfortunately, this does not happen without undesirable "side-effects" because derivation generates a "loss of details". Let's look at the example 5.3.

Example 5.3.

Let m_1 and m_2 be two messages of a generalized role of a protocol p such that $m_1 = \{\alpha.C.X\}_{k_{ab}}$ and $m_2 = \{\alpha.Y.D\}_{k_{ab}}$ and $\ulcorner \alpha \urcorner = \{A, B\}$;

Let $m = \{\alpha.C.D\}_{k_{as}}$ be a closed message in a valid trace generated by p ;

$$F_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]m) = \begin{cases} \{C, A, B\} & \text{if } m \text{ comes by the substitution of } X \text{ by } D \text{ in } m_1 \\ \{D, A, B\} & \text{if } m \text{ comes by the substitution of } Y \text{ by } C \text{ in } m_2 \end{cases}$$

Hence $F_{MAX}^{EK}(\alpha, \partial[\overline{\alpha}]m)$ is not even a function on the closed message m since it may return more than one image for the same preimage. This leads us straightly to the Witness-Functions.

6 The Witness-Functions

Definition 6.1. (Witness-Function)

Let $m \in \mathcal{M}_p^G$, $X \in \mathcal{X}_m$ and $m\sigma$ be a closed message.

Let p be a protocol and F be a \mathcal{C} -reliable function.

We define a Witness-Function $\mathcal{W}_{p,F}$ for all $\alpha \in \mathcal{A}(m\sigma)$, $\sigma \in \Gamma$, as follows:

$$\mathcal{W}_{p,F}(\alpha, m\sigma) = \bigcap_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = m\sigma}} F(\alpha, \partial[\bar{\alpha}]m'\sigma')$$

$\mathcal{W}_{p,F}$ is said to be a Witness-Function inside the protection of an external key when F is a function such that its associated selection is an instance of the class S_{Gen}^{EK} .

According to the example 5.3, the application defined in 5.2 is not necessary a function in \mathcal{M}_p^G as a valid trace could have more than one source (or provenance) in \mathcal{M}_p^G and each source has a different static part. A Witness-Function is yet a function in \mathcal{M}_p^G since it searches all the sources of the closed message in input and returns the minimum (the union). This minimum naturally exists and is unique in the finite set \mathcal{M}_p^G . A Witness-Function is protocol-dependent as it depends on messages in the generalized roles of the protocol. However, it is built uniformly for any pair (protocol, function) in input. For a Witness-Function inside the protection of an external key, since its associated function calculates the level of security of an atom always in a message m having an encryption pattern, the search of the sources of the closed message $m\sigma$ in \mathcal{M}_p^G (i.e. $\{m' \in \mathcal{M}_p^G \mid \exists \sigma' \in \Gamma. m'\sigma' = m\sigma\}$) is reduced to a search in the encryption patterns in \mathcal{M}_p^G .

6.1 Legacy of Reliability

Proposition 6.2 asserts that a function F inside the protection of an external key transmits its reliability to its associated Witness-Function $\mathcal{W}_{p,F}$. In fact, the selection associated with a Witness-Function inside the protection of an external key is the union of selections associated with the function F , limited to derivative messages. It is easy to check that a Witness-Function is well-formed. Concerning the full-invariance-by-intruder property, as the derivation just eliminates variables (so some atoms when the message is substituted), and since each selection in the union returns a subset among allowed candidates, then the union itself returns a subset among allowed candidates (the union of subsets is a subset). Therefore, the selection associated with a Witness-Function stays an instance of the class S_{Gen}^{EK} , so full-invariant-by-intruder. Since the Witness-Function is the composition of the homomorphism of F and an instance of the class S_{Gen}^{EK} , then it is reliable.

Proposition 6.2.

Let $\mathcal{W}_{p,F}$ be a Witness-Function inside the protection of an external key.

We have:

$$\mathcal{W}_{p,F} \text{ inherits reliability from } F$$

See the proof 18 in [14]

Example 6.3.

Let $\mathcal{M}_p^G = \{\{\alpha.B.X\}_{k_{ad}}, \{\alpha.Y.C\}_{k_{ad}}, \{A.Z\}_{k_{bc}}\}$;

$Var(\mathcal{M}_p^G) = \{X, Y, Z\}$; $m_1 = \{\alpha.B.C\}_{k_{ad}}$; $\ulcorner \alpha \urcorner = \{A, D\}$; $\ulcorner k_{ad}^{-1} \urcorner = \{A, D\}$;

$\mathcal{W}_{p,F_{MAX}^{EK}}(\alpha, m_1)$

= {Definition 6.1}

$$\bigcap_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = m_1}} F_{MAX}^{EK}(\alpha, \partial[\bar{\alpha}]m'\sigma') = \bigcap_{\substack{\{\{\alpha.B.X\}_{k_{ad}}, \{\alpha.Y.C\}_{k_{ad}}\} \\ \sigma' = \{X \mapsto C, Y \mapsto B\}}} F_{MAX}^{EK}(\alpha, \partial[\bar{\alpha}]m'\sigma')$$

= $\{\mathcal{W}_{p,F_{MAX}^{EK}} \text{ is well-formed from Proposition 6.2}\}$

$$F_{MAX}^{EK}(\alpha, \partial[\bar{\alpha}]\{\alpha.B.X\}_{k_{ad}}[X \mapsto C]) \sqcap F_{MAX}^{EK}(\alpha, \partial[\bar{\alpha}]\{\alpha.Y.C\}_{k_{ad}}[Y \mapsto B])$$

= {Definition 5.2 and derivation in 5.1}

$$F_{MAX}^{EK}(\alpha, \{\alpha.B\}_{k_{ad}}) \sqcap F_{MAX}^{EK}(\alpha, \{\alpha.C\}_{k_{ad}})$$

= {Definition of F_{MAX}^{EK} }

$$\{B, A, D\} \cup \{C, A, D\} = \{B, A, D, C\}$$

6.2 Bounds of a Witness-Function

In Lemma 6.4, we define two interesting bounds of a Witness-Function that are independent of all substitutions. The upper bound of a Witness-Function ranks the security level of an atom α in a closed message $m\sigma$ from one *confirmed* source m (m is a natural source of $m\sigma$), the Witness-Function itself ranks it from the *exact* sources of $m\sigma$ that are known only when the protocol is run, and the lower bound ranks it from all *likely* sources of $m\sigma$ (i.e. the messages that are unifiable with m in \mathcal{M}_p^G). The unification in the lower bound catches all the principal identities that could participate in a possible intrusion. It is worth mentioning that the upper bound and the lower bound of the Witness-Function $\mathcal{W}_{p,F}$ do not depend on the substitution σ that generates the closed message $m\sigma$.

Lemma 6.4.

Let $m \in \mathcal{M}_p^G$ and $\mathcal{W}_{p,F}$ be a Witness-Function inside the protection of an external key.
 $\forall \sigma \in \Gamma, \forall \alpha \in \mathcal{A}(\mathcal{M}_p)$ we have:

$$F(\alpha, \partial[\bar{\alpha}]m) \sqsupseteq \mathcal{W}_{p,F}(\alpha, m\sigma) \sqsupseteq \bigcap_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = m\sigma}} F(\alpha, \partial[\bar{\alpha}]m'\sigma')$$

Proof.

For any $\sigma \in \Gamma$ we have:

- $F(\alpha, \partial[\bar{\alpha}]m) \sqsubseteq \mathcal{W}_{p,F}(\alpha, m\sigma)$: since m is obviously one element of the set $\{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m'\sigma' = m\sigma\}$ of calculation of $\mathcal{W}_{p,F}(\alpha, m\sigma)$ (i.e. m is a trivial source of $m\sigma$) and since $F(\alpha, \partial[\bar{\alpha}]m\sigma)$ does not depend on σ because, by construction, it depends only on the static part of m (denoted simply by $F(\alpha, \partial[\bar{\alpha}]m)$).
- $\mathcal{W}_{p,F}(\alpha, m\sigma) \sqsubseteq \bigcup_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = m\sigma}} F(\alpha, \partial[\bar{\alpha}]m'\sigma')$: since for all $m \in \mathcal{M}_p^G$ the set $\{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m'\sigma' = m\sigma\}$ (unifications) is obviously larger than the set $\{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m'\sigma' = m\sigma\}$ of sources of $m\sigma$ in \mathcal{M}_p^G .

From these two facts and since \mathcal{L}^\sqsupseteq is a lattice, we have the result in Lemma 6.4

7 A SEMI-DECIDABLE PROCEDURE FOR ANALYZING CRYPTOGRAPHIC PROTOCOLS WITH THE WITNESS-FUNCTIONS

Now, we give the protocol analysis with a Witness-Function theorem that sets a criterion for protocols correctness with respect to the secrecy property. The result in Theorem 7.1 derives directly from Proposition 6.2, Lemma 6.4 and Theorem 3.7.

Theorem 7.1. (Protocol analysis with a Witness-Function)

Let $\mathcal{W}_{p,F}$ be a Witness-Function inside the protection of an external key.

A sufficient condition of correctness of p with respect to the secrecy property is:

$\forall R. r \in R_G(p), \forall \alpha \in \mathcal{A}(r^+)$ we have:

$$\bigcap_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = r^+\sigma'}} F(\alpha, \partial[\bar{\alpha}]m'\sigma') \sqsupseteq \ulcorner \alpha \urcorner \sqcap F(\alpha, \partial[\bar{\alpha}]R^-)$$

Proof.

Suppose we have: $\forall R. r \in R_G(p), \forall \alpha \in \mathcal{A}(r^+)$

$$\bigcap_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = r^+\sigma'}} F(\alpha, \partial[\bar{\alpha}]m'\sigma') \sqsupseteq \ulcorner \alpha \urcorner \sqcap F(\alpha, \partial[\bar{\alpha}]R^-) \quad (14)$$

From Lemma 6.4 and since \mathcal{L}^\sqsupseteq is a lattice we have for all $\sigma \in \Gamma$:

$$\forall \alpha \in \mathcal{A}(\mathcal{M}_p). \mathcal{W}_{p,F}(\alpha, r^+\sigma) \sqsupseteq \bigcap_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = r^+\sigma'}} F(\alpha, \partial[\bar{\alpha}]m'\sigma') \quad (15)$$

$$\text{and } \forall \alpha \in \mathcal{A}(\mathcal{M}_p). \ulcorner \alpha \urcorner \sqcap F(\alpha, \partial[\bar{\alpha}]R^-) \sqsupseteq \ulcorner \alpha \urcorner \sqcap \mathcal{W}_{p,F}(\alpha, R^-\sigma) \quad (16)$$

From 14, 15 and 16 we have:

$$\forall \alpha \in \mathcal{A}(r^+ \sigma). \mathcal{W}_{p,F}(\alpha, r^+ \sigma) \sqsupseteq \ulcorner \alpha \urcorner \sqcap \mathcal{W}_{p,F}(\alpha, R^- \sigma) \quad (17)$$

From Proposition 6.2 $\mathcal{W}_{p,F}$ is \mathcal{C} -reliable, then we have from Theorem 3.1 and 17:

p is correct with respect to the secrecy property \square

The secrecy in cryptographic protocols is in general an undecidable problem [19–23]. Theorem 7.1 defines a semi-decidable procedure for analyzing cryptographic protocols with the Witness-Functions for the class of increasing protocols. Hence, the problem of protocol correctness becomes a problem of protocol growth. If the criterion in Theorem 7.1 is met, we can conclude that the protocol is correct for secrecy. Otherwise, no decision can be made. This criterion uses the upper bound and the lower bound of a Witness-Function. The upper bound gives a reasonable estimation of the level of security of an atom in a message on its reception. This estimation is based on selection among the external protective key and the principal identities that travel with it under the same protection. The lower bound makes sure that no unauthorized principal identity is maliciously inserted beside this atom on sending and therefore ensures that no intrusion could happen. The fact that the two bounds of the Witness-Function are not affected by substitution, our semi-decidable procedure can transfer any decision made on the generalized roles of a protocol to the valid traces and with no restriction on their size.

If an analysis with a Witness-Function fails to prove the correctness of a protocol for secrecy, we can change the Witness-Function by changing its function in entry. We can also have an idea about some flaw in it. If all the Witness-Functions that we manage to build fail, we can advice an amendment to the structure of the protocol by changing some encryption keys or principal identities in the exchanged messages in the protocol. The procedure steps are given in Figure 1.

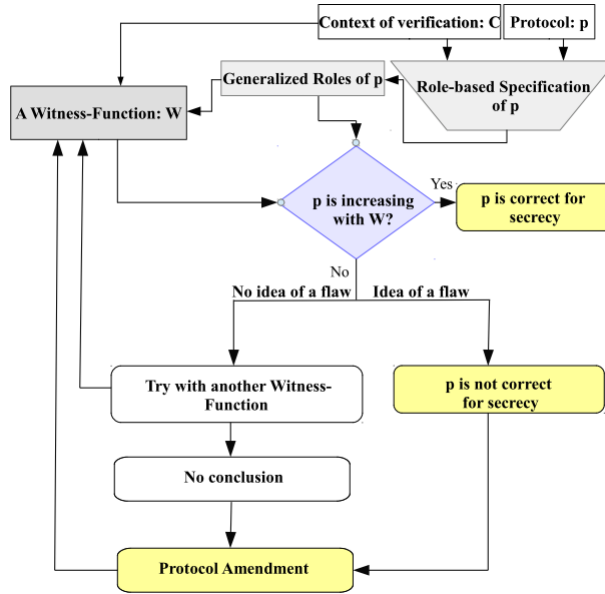


Figure 1: Analyzing cryptographic protocols with Witness-Functions

8 ANALYSIS OF THE NEEDHAM-SCHROEDER PROTOCOL WITH A WITNESS-FUNCTION

In this section, we analyze the Needham-Schroeder protocol given in Example 2.2 with a Witness-Function. Let us have a context of verification such that: $\ulcorner A \urcorner = \perp$; $\ulcorner B \urcorner = \perp$; $\ulcorner N_a^i \urcorner = \{A, B\}$ (secret between A and B); $\ulcorner N_b^i \urcorner = \{A, B\}$ (secret between A and B); $\ulcorner k_a^{-1} \urcorner = \{A\}$; $\ulcorner k_b^{-1} \urcorner = \{B\}$; $(\mathcal{L}, \sqsupseteq, \sqcup, \sqcap, \perp, \top) = (2^{\mathcal{I}}, \subseteq, \cap, \cup, \mathcal{I}, \emptyset)$; $\mathcal{I} = \{I(\text{intruder}), A, B, C, A_1, A_2, B_1, B_2, \dots\}$;

The set of messages generated by the protocol is $\mathcal{M}_p^G = \{\{A_1.N_{A_1}\}_{k_{B_1}}, \{N_{A_2}.X_1.B_2\}_{k_{A_2}}, \{X_2\}_{k_{B_3}}, \{A_3.Y_1\}_{k_{B_4}}, \{Y_2.N_{B_5}.B_5\}_{k_{A_4}}, \{N_{B_6}\}_{k_{B_6}}\}$;

The variables are denoted by X_1, X_2, Y_1 and Y_2 ;

The static names are denoted by $A_1, N_{A_1}, k_{B_1}, N_{A_2}, B_2, k_{A_2}, k_{B_3}, A_3, k_{B_4}, N_{B_5}, B_5, k_{A_4}, N_{B_6}$ and k_{B_6} ;

Let's select the Witness-Function as follows:

$$p = NS; F = F_{MAX}^{EK}; \mathcal{W}_{p,F}(\alpha, m\sigma) = \bigcap_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = m\sigma}} F(\alpha, \partial[\bar{\alpha}]m'\sigma');$$

Let's denote the lower bound of the Witness-Function by:

$$\mathcal{W}'_{p,F}(\alpha, r^+) = \bigcap_{\substack{m' \in \mathcal{M}_p^G \\ \exists \sigma' \in \Gamma. m'\sigma' = r^+\sigma'}} F(\alpha, \partial[\bar{\alpha}]m'\sigma')$$

8.0.1 Analysis of the generalized role of A

As defined in the generalized roles of p , an agent A can participate in two consequent sessions: S^i and S^j such that $j > i$. In the former session S^i , the agent A receives nothing and sends the message $\{A.N_a^i\}_{k_b}$. In a consequent session S^j , she receives the message $\{N_a^i.X.B\}_{k_a}$ and she sends the message $\{X\}_{k_b}$. This is described by the following schema:

$$S^i : \frac{\square}{\{A.N_a^i\}_{k_b}} \quad S^j : \frac{\{N_a^i.X.B\}_{k_a}}{\{X\}_{k_b}}$$

-Analysis of the messages exchanged in the session S^i :

1- For any N_a^i :

a- When sending: $r_{S^i}^+ = \{A.N_a^i\}_{k_b}$ (in a sending step, the lower bound is used)

$$\forall N_a^i. \{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m'\sigma' = r_{S^i}^+\sigma'\}$$

$$= \forall N_a^i. \{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m'\sigma' = \{A.N_a^i\}_{k_b}\sigma'\}$$

$$= \{(\{A_1.N_{A_1}\}_{k_{B_1}}, \sigma'_1), (\{X_2\}_{k_{B_3}}, \sigma'_2), (\{A_3.Y_1\}_{k_{B_4}}, \sigma'_3)\} \text{ such that:}$$

$$\begin{cases} \sigma'_1 = \{A_1 \mapsto A, N_{A_1} \mapsto N_a^i, k_{B_1} \mapsto k_b\} \\ \sigma'_2 = \{X_2 \mapsto A.N_a^i, k_{B_3} \mapsto k_b\} \\ \sigma'_3 = \{A_3 \mapsto A, Y_1 \mapsto N_a^i, k_{B_4} \mapsto k_b\} \end{cases}$$

$$\mathcal{W}'_{p,F}(N_a^i, \{A.N_a^i\}_{k_b})$$

$$= \{\text{Definition of the lower bound of the Witness-Function}\}$$

$$F(N_a^i, \partial[\bar{N}_a^i]\{A_1.N_{A_1}\}_{k_{B_1}}\sigma'_1) \sqcap F(N_a^i, \partial[\bar{N}_a^i]\{X_2\}_{k_{B_3}}\sigma'_2) \sqcap F(N_a^i, \partial[\bar{N}_a^i]\{A_3.Y_1\}_{k_{B_4}}\sigma'_3)$$

$$= \{\text{Setting the static neighborhood}\}$$

$$F(N_a^i, \partial[\bar{N}_a^i]\{A.N_a^i\}_{k_b}\sigma'_1) \sqcap F(N_a^i, \partial[\bar{N}_a^i]\{X_2\}_{k_b}\sigma'_2) \sqcap F(N_a^i, \partial[\bar{N}_a^i]\{A.Y_1\}_{k_b}\sigma'_3)$$

$$= \{\text{Definition 5.2}\}$$

$$F(N_a^i, \{A.N_a^i\}_{k_b}) \sqcap F(X_2, \partial[\bar{X}_2]\{X_2\}_{k_b}) \sqcap F(Y_1, \partial[\bar{Y}_1]\{A.Y_1\}_{k_b})$$

$$= \{\text{Derivation in Definition 5.1}\}$$

$$F(N_a^i, \{A.N_a^i\}_{k_b}) \sqcap F(X_2, \{X_2\}_{k_b}) \sqcap F(Y_1, \{A.Y_1\}_{k_b})$$

$$= \{\text{Since } F = F_{MAX}^{EK}\}$$

$$\{A, B\} \cup \{B\} \cup \{A, B\} = \{A, B\} \quad (1.0)$$

b- When receiving: $R_{S^i}^- = \emptyset$ (in a receiving step, the upper bound is used)

$$F(N_a^i, \emptyset) = \top \quad (1.1)$$

2- Conformity to the criterion set by Theorem 7.1 of the exchanged messages in the session S^i :

From (1.0) and (1.1), we have:

$$\mathcal{W}'_{p,F}(N_a^i, \{A.N_a^i\}_{k_b}) = \{A, B\} \sqsupseteq \top N_a^i \sqcap F(N_a^i, \emptyset) = \{A, B\} \sqcap \top = \{A, B\} \quad (1.2)$$

From (1.2), the messages exchanged in the session S^i respect the criterion set by Theorem 7.1. (I)

-Analysis of the messages exchanged in the session S^j :

1- For any X :

a- When sending: $R_{S^j}^+ = \{X\}_{k_b}$ (in a sending step, the lower bound is used)

$$\begin{aligned} & \forall X. \{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m' \sigma' = R_{S^j}^+ \sigma'\} = \forall X. \{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m' \sigma' = \{X\}_{k_b} \sigma'\} \\ & = \{(\{X_2\}_{k_{B_3}}, \sigma'_1)\} \text{ such that: } \sigma'_1 = \{X_2 \mapsto X, k_{B_3} \mapsto k_b\}. \end{aligned}$$

$$\begin{aligned} & \mathcal{W}'_{p,F}(X, \{X\}_{k_b}) \\ & = \{\text{Definition of the lower bound of the Witness-Function}\} \\ & F(X, \partial[\overline{X}]\{X_2\}_{k_{B_3}} \sigma'_1) \\ & = \{\text{Setting the static neighborhood}\} \\ & F(X, \partial[\overline{X}]\{X_2\}_{k_b} \sigma'_1) \\ & = \{\text{Definition 5.2}\} \\ & F(X_2, \partial[\overline{X_2}]\{X_2\}_{k_b}) \\ & = \{\text{Derivation in Definition 5.1}\} \\ & F(X_2, \{X_2\}_{k_b}) \\ & = \{\text{Since } F = F_{MAX}^{EK}\} \\ & \{B\} \text{ (2.0)} \end{aligned}$$

b- When receiving: $R_{S^j}^- = \{N_a^i.X.B\}_{k_a}$ (in a receiving step, the upper bound is used)

$$F(X, \partial[\overline{X}]\{N_a^i.X.B\}_{k_a}) = \{A, B\} \text{ (2.1)}$$

3- Conformity to the criterion set by Theorem 7.1 of the exchanged messages in the session S^j :

From (2.0) et (2.1), we have:

$$\mathcal{W}'_{p,F}(X, \{X\}_{k_b}) = \{B\} \sqsupseteq \ulcorner X \urcorner \sqcap F(X, \partial[\overline{X}]\{N_a^i.X.B\}_{k_a}) = \ulcorner X \urcorner \sqcup \{A, B\} \text{ (2.2)}$$

From (2.2), the messages exchanged in the session S^j respect the criterion set by Theorem 7.1. (II)

From (I) and (II), the messages exchanged in the generalized role of A respect the criterion set by Theorem 7.1. (III)

8.0.2 Analysis of the generalized role of B

As defined in the generalized roles of p , an agent B can participate in a session S'^i , in which she receives the message $\{A.Y\}_{k_b}$ and she sends the message $\{Y.N_b^i.B\}_{k_a}$. This is described by the following schema:

$$S'^i : \frac{\{A.Y\}_{k_b}}{\{Y.N_b^i.B\}_{k_a}}$$

-Analysis of the messages exchanged in the session S'^i :

1- For any Y :

a- When sending: $r_{S'^i}^+ = \{Y.N_b^i.B\}_{k_a}$ (in a sending step, the lower bound is used)

$$\begin{aligned} & \forall Y. \{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m' \sigma' = r_{S'^i}^+ \sigma'\} \\ & = \forall Y. \{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m' \sigma' = \{Y.N_b^i.B\}_{k_a} \sigma'\} \\ & = \{(\{Y_2.N_{B_5}.B_5\}_{k_{A_4}}, \sigma'_1), (\{X_2\}_{k_{B_3}}, \sigma'_2)\} \text{ such that:} \end{aligned}$$

$$\begin{cases} \sigma'_1 = \{Y_2 \mapsto Y, N_{B_5} \mapsto N_b^i, B_5 \mapsto B, k_{A_4} \mapsto k_a\} \\ \sigma'_2 = \{X_2 \mapsto Y.N_b^i.B, k_{B_3} \mapsto k_a\} \end{cases}$$

$$\begin{aligned} & \mathcal{W}'_{p,F}(Y, \{Y.N_b^i.B\}_{k_a}) \\ & = \{\text{Definition of the lower bound of the Witness-Function}\} \\ & F(Y, \partial[\overline{Y}]\{Y_2.N_{B_5}.B_5\}_{k_{A_4}} \sigma'_1) \sqcap F(Y, \partial[\overline{Y}]\{X_2\}_{k_{B_3}} \sigma'_2) \\ & = \{\text{Setting the static neighborhood}\} \\ & F(Y, \partial[\overline{Y}]\{Y_2.N_b^i.B\}_{k_a} \sigma'_1) \sqcap F(Y, \partial[\overline{Y}]\{X_2\}_{k_a} \sigma'_2) \\ & = \{\text{Definition 5.2}\} \end{aligned}$$

$$\begin{aligned}
& F(Y_2, \partial[\overline{Y_2}]\{Y_2.N_b^i.B\}_{k_a}) \sqcap F(X_2, \partial[\overline{X_2}]\{X_2\}_{k_a}) \\
& = \{\text{Derivation in Definition 5.1}\} \\
& F(Y_2, \{Y_2.N_b^i.B\}_{k_a}) \sqcap F(X_2, \{X_2\}_{k_a}) \\
& = \{\text{Since } F = F_{MAX}^{EK}\} \\
& \{A, B\} \cup \{A\} = \{A, B\} \quad (3.0)
\end{aligned}$$

b- When receiving: $R_{S'^i}^- = \{A.Y\}_{k_b}$ (in a receiving step, the upper bound is used)

$$F(Y, \partial[\overline{Y}]\{A.Y\}_{k_b}) = \{A, B\} \quad (3.1)$$

2- For any N_b^i :

a- When sending: $r_{S'^i}^+ = \{Y.N_b^i.B\}_{k_a}$ (in a sending step, the lower bound is used)

$$\begin{aligned}
& \forall N_b^i. \{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m' \sigma' = r_{S'^i}^+ \sigma'\} \\
& = \forall N_b^i. \{m' \in \mathcal{M}_p^G | \exists \sigma' \in \Gamma. m' \sigma' = \{Y.N_b^i.B\}_{k_a} \sigma'\} \\
& = \{(\{Y_2.N_{B_5}.B_5\}_{k_{A_4}}, \sigma'_1), (\{X_2\}_{k_{B_3}}, \sigma'_2), (\{A_3.Y_1\}_{k_{B_4}}, \sigma'_3)\} \text{ such that:}
\end{aligned}$$

$$\begin{cases} \sigma'_1 = \{Y_2 \mapsto Y, N_{B_5} \mapsto N_b^i, B_5 \mapsto B, k_{A_4} \mapsto k_a\} \\ \sigma'_2 = \{X_2 \mapsto Y.N_b^i.B, k_{B_3} \mapsto k_a\} \\ \sigma'_3 = \{Y \mapsto A_3, Y_1 \mapsto N_b^i.B, k_{B_4} \mapsto k_a\} \end{cases}$$

$$\begin{aligned}
& \mathcal{W}'_{p,F}(N_b^i, \{Y.N_b^i.B\}_{k_a}) \\
& = \{\text{Definition of the lower bound of the Witness-Function}\} \\
& F(N_b^i, \partial[\overline{N_b^i}]\{Y_2.N_{B_5}.B_5\}_{k_{A_4}} \sigma'_1) \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{X_2\}_{k_{B_3}} \sigma'_2) \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{A_3.Y_1\}_{k_{B_4}} \sigma'_3) \\
& = \{\text{Setting the static neighborhood}\} \\
& F(N_b^i, \partial[\overline{N_b^i}]\{Y_2.N_b^i.B\}_{k_a} \sigma'_1) \sqcap \\
& F(N_b^i, \partial[\overline{N_b^i}]\{X_2\}_{k_a} \sigma'_2) \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{A_3.Y_1\}_{k_a} \sigma'_3) \\
& = \{\text{Definition 5.2}\} \\
& F(N_b^i, \{N_b^i.B\}_{k_a}) \sqcap F(X_2, \partial[\overline{X_2}]\{X_2\}_{k_a}) \sqcap F(Y_1, \partial[\overline{Y_1}]\{A_3.Y_1\}_{k_a}) \\
& = \{\text{Derivation in Definition 5.1}\} \\
& F(N_b^i, \{N_b^i.B\}_{k_a}) \sqcap F(X_2, \{X_2\}_{k_a}) \sqcap F(Y_1, \{A_3.Y_1\}_{k_a}) \\
& = \{\text{Since } F = F_{MAX}^{EK}\} \\
& \{A, B\} \cup \{A\} \cup \{A_3, A\} = \{A, B, A_3\} \quad (3.2)
\end{aligned}$$

b- When receiving: $R_{S'^i}^- = \{A.Y\}_{k_b}$ (in a receiving step, the upper bound is used)

$$F(N_b^i, \partial[\overline{N_b^i}]\{A.Y\}_{k_b}) = \top \quad (3.3)$$

3- Conformity to the criterion set by Theorem 7.1 of the exchanged messages in the session S'^i :

From (3.0) et (3.1), we have:

$$\mathcal{W}'_{p,F}(Y, \{Y.N_b^i.B\}_{k_a}) = \{A, B\} \sqsubseteq \lceil Y \rceil \sqcap F(Y, \partial[\overline{Y}]\{A.Y\}_{k_b}) = \lceil Y \rceil \cup \{A, B\} \quad (3.4)$$

From (3.2) et (3.3), we have:

$$\begin{aligned}
& \mathcal{W}'_{p,F}(N_b^i, \{Y.N_b^i.B\}_{k_a}) = \{A, B, A_3\} \not\sqsubseteq \lceil N_b^i \rceil \sqcap F(N_b^i, \partial[\overline{N_b^i}]\{A.Y\}_{k_b}) = \lceil N_b^i \rceil \sqcap \top = \lceil N_b^i \rceil = \{A, B\} \\
& (3.5). \text{ From (3.5), the messages exchanged in the session } S'^i \text{ do not respect the criterion set by Theorem} \\
& 7.1. \text{ (IV)}
\end{aligned}$$

8.1 Results and Interpretation

The results of the analysis of the variation of Needham-Schroeder protocol are summarized in Table 3.

We notice from Table 3 that the variation of Needham-Schroeder protocol does not respect the criterion set by Theorem 7.1 when analyzed with the witness-function $\mathcal{W}_{p_{NS}, F_{MAX}^{EK}}$. Therefore, we cannot deduce anything regarding its correctness with respect to the secrecy property. The non-growth of the protocol is localized in the sending step of the generalized role of B and it is due to a possible malicious neighbor (denoted by A_3 in our analysis) that could be inserted beside the nonce N_B^i . In the literature, we report a flaw that operates on the decay of the level of security of the nonce N_B^i in the generalized role of B . This flaw is described by the attack scenario in Figure.2.

Table 3: Conformity of the Needham-Schroeder Protocol to Theorem 7.1

α	Role	R^-	r^+	$\mathcal{W}'_{p,F}(\alpha, r^+)$	$\lceil \alpha \rceil$	$F(\alpha, \partial[\bar{\alpha}]R^-)$	Theorem 7.1
N_a^i	A	\emptyset	$\{A.N_a^i\}_{k_b}$	$\{A, B\}$	$\{A, B\}$	\top	Fulfilled
$\forall X$	A	$\{N_a^i.X.B\}_{k_a}$	$\{X\}_{k_b}$	$\{B\}$	$\lceil X \rceil$	$\{A, B\}$	Fulfilled
$\forall Y$	B	$\{A.Y\}_{k_b}$	$\{Y.N_b^i.B\}_{k_a}$	$\{A, B\}$	$\lceil Y \rceil$	$\{A, B\}$	Fulfilled
N_b^i	B	$\{A.Y\}_{k_b}$	$\{Y.N_b^i.B\}_{k_a}$	$\{A, B, A_3\}$	$\{A, B\}$	$\{A, B\}$	Not Fulfilled

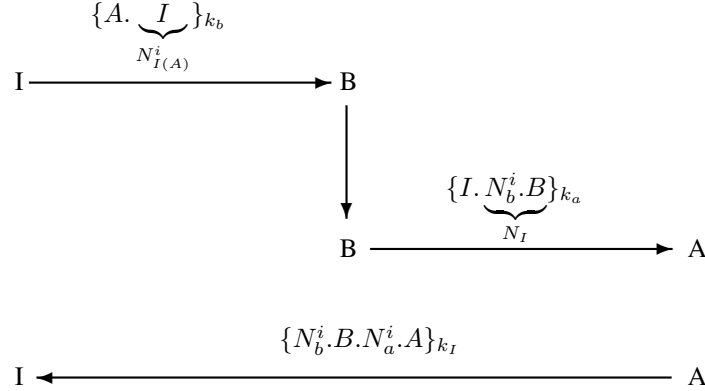


Figure 2: Attack Scenario on the Needham-Schroeder Protocol

9 ANALYSIS OF THE NSL PROTOCOL WITH A WITNESS-FUNCTION

In this section, we analyze the NSL protocol with a witness-function given in Example 2.3. Let's have a context of verification such that: $\lceil A \rceil = \perp$; $\lceil B \rceil = \perp$; $\lceil N_a^i \rceil = \{A, B\}$; $\lceil N_b^i \rceil = \{A, B\}$; $\lceil k_a^{-1} \rceil = \{A\}$; $\lceil k_b^{-1} \rceil = \{B\}$; $(\mathcal{L}, \sqsubseteq, \sqcup, \sqcap, \perp, \top) = (2^{\mathcal{I}}, \subseteq, \cap, \cup, \mathcal{I}, \emptyset)$; $\mathcal{I} = \{I, A, B, A_1, A_2, B_1, B_2, \dots\}$; The set of messages generated by the protocol is $\mathcal{M}_p^G = \{\{N_{A_1}.A_1\}_{k_{B_1}}, \{B_2.N_{A_2}\}_{k_{A_2}}, \{B_3.X_1\}_{k_{A_3}}, \{X_2\}_{k_{B_4}}, \{Y_1.A_4\}_{k_{B_5}}, \{B_7.N_{B_7}\}_{k_{A_6}}, \{N_{B_8}\}_{k_{B_8}}\}$. The variables are denoted by X_1, X_2, Y_1 and Y_2 ; The static names are denoted by $N_{A_1}, A_1, k_{B_1}, B_2, N_{A_2}, k_{A_2}, B_3, k_{A_3}, k_{B_4}, A_4, k_{B_5}, B_7, N_{B_7}, k_{A_6}, N_{B_8}$ and k_{B_8} .

9.1 Results and Interpretation

The results of analysis of the NSL protocol are summarized in Table 4. We notice from Table 4 that the

Table 4: Conformity of NSL Protocol to Theorem 7.1

α	Role	R^-	r^+	$\mathcal{W}'_{p,F}(\alpha, r^+)$	$\lceil \alpha \rceil$	$F(\alpha, \partial[\bar{\alpha}]R^-)$	Theorem 7.1
N_a^i	A	\emptyset	$\{A.N_a^i\}_{k_b}$	$\{A, B\}$	$\{A, B\}$	\top	Fulfilled
$\forall X$	A	$\{B.N_a^i\}_{k_a}.\{B.X\}_{k_a}$	$A.B.\{X\}_{k_b}$	$\{B\}$	$\lceil X \rceil$	$\{A, B\}$	Fulfilled
$\forall Y$	B	$\{A.Y\}_{k_b}$	$\{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}$	$\{A, B\}$	$\lceil Y \rceil$	$\{A, B\}$	Fulfilled
N_b^i	B	$\{A.Y\}_{k_b}$	$\{B.Y\}_{k_a}.\{B.N_b^i\}_{k_a}$	$\{A, B\}$	$\{A, B\}$	$\{A, B\}$	Fulfilled

NSL protocol respects the correctness criterion stated in Theorem 7.1, therefore it is correct with respect to the secrecy property.

10 COMPARISON WITH RELATED WORKS

Houmani et al. in [5–8] defined universal functions to analyze protocols. Although they gave a comprehensive guideline to build these functions, only two functions had been concretely defined: DEK and DEKAN. That is due to the difficulty to find, and after to prove, that a function satisfies to the property of full-invariance by substitution. With a Witness-Function, there is no need to this property. A Witness-Function operates on derivative messages instead of messages of the generalized roles of a protocol which reduces variables' effect. It also offers two helpful bounds that do not depend on substitutions. This relaxes the conditions made on the associated interpretation functions and allows us to have more reliable functions and therefore to prove secrecy in a wider range of protocols. In addition, our method does not need any complicated context as do the rank-functions of Schneider in [1] that require the CSP formalism [24]. It does not need neither a strong message-typing as required by Abadi in [25]. Our method joins though Houmani's method and Schneider's method in the way of seeing secrecy in a protocol through its growth.

11 CONCLUSION AND FUTURE WORK

In this paper, we suggested a new semi-decidable procedure to analyze cryptographic protocols with the Witness-Functions based on selection inside the external protective key for the property of secrecy. We experimented our approach on real protocols and we showed that the Witness-functions can even help to teach about flaws. They were successful to prove secrecy in others too. As a future work, we intend to define more Witness-Functions based on selection inside other keys like the most internal key or a group of keys together to enlarge the range of protocols that could be proved correct. We intend also to check our approach on protocols with non-empty theories [26–28]. In addition, we intend to take advantage of the bounds offered by Witness-Functions so that we can use exchanged messages in a protocol as encryption keys. We think also to extend our approach so that it can deal with other security properties like authentication and integrity.

References

- [1] S. Schneider, "Verifying authentication protocols in csp," *IEEE Trans. Software Eng.*, vol. 24, no. 9, pp. 741–758, 1998.
- [2] S. Schneider, "Security properties and csp," in *IEEE Symposium on Security and Privacy*, pp. 174–187, 1996.
- [3] S. A. Schneider and R. Delicata, "Verifying security protocols: An application of csp," in *25 Years Communicating Sequential Processes*, pp. 243–263, 2004.
- [4] J. Heather and S. Schneider, "A decision procedure for the existence of a rank function," *J. Comput. Secur.*, vol. 13, pp. 317–344, Mar. 2005.
- [5] H. Houmani and M. Mejri, "Practical and universal interpretation functions for secrecy," in *SECRYPT*, pp. 157–164, 2007.
- [6] H. Houmani and M. Mejri, "Ensuring the correctness of cryptographic protocols with respect to secrecy," in *SECRYPT*, pp. 184–189, 2008.
- [7] H. Houmani and M. Mejri, "Formal analysis of set and nsl protocols using the interpretation functions-based method," *Journal Comp. Netw. and Communic.*, vol. 2012, 2012.
- [8] H. Houmani, M. Mejri, and H. Fujita, "Secrecy of cryptographic protocols under equational theory," *Knowl.-Based Syst.*, vol. 22, no. 3, pp. 160–173, 2009.
- [9] D. Dolev and A. C.-C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–207, 1983.
- [10] J. Fattahi, M. Mejri, and H. Houmani, "Context of verification and role-based specification http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Ind/Context.pdf," pp. 1–4, 2014.
- [11] M. Debbabi, Y. Legaré, and M. Mejri, "An environment for the specification and analysis of cryptoprotocols," in *ACSAC*, pp. 321–332, 1998.
- [12] M. Debbabi, M. Mejri, N. Tawbi, and I. Yahmadi, "Formal automatic verification of authentication cryptographic protocols," in *ICFEM*, pp. 50–59, 1997.
- [13] M. Debbabi, M. Mejri, N. Tawbi, and I. Yahmadi, "From protocol specifications to flaws and attack scenarios: An automatic and formal algorithm," in *WETICE*, pp. 256–262, 1997.

- [14] J. Fattahi, M. Mejri, and H. Houmani, "The witness-functions: Proofs and intermediate results. http://web_security.fsg.ulaval.ca/lab/sites/default/files/WF/Ind/WitFunProofsJ.pdf," pp. 1–26, 2014.
- [15] B. Blanchet, "Automatic verification of correspondences for security protocols," *Journal of Computer Security*, vol. 17, no. 4, pp. 363–434, 2009.
- [16] F. Baader and T. Nipkow, *Term rewriting and all that*. Cambridge University Press, 1998.
- [17] N. Dershowitz and D. A. Plaisted, "Rewriting," in *Handbook of Automated Reasoning*, pp. 535–610, 2001.
- [18] H. Comon-Lundh, C. Kirchner, and H. Kirchner, eds., *Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday*, vol. 4600 of *Lecture Notes in Computer Science*, Springer, 2007.
- [19] V. Cortier and S. Delaune, "Decidability and combination results for two notions of knowledge in security protocols," *Journal of Automated Reasoning*, vol. 48, 2012.
- [20] V. Cortier, S. Delaune, and P. Lafourcade, "A survey of algebraic properties used in cryptographic protocols," *J. Comput. Secur.*, vol. 14, pp. 1–43, Jan. 2006.
- [21] J. M. N. Durgin, P. Lincoln and A. Scedrov., "Undecidability of bounded security protocols," *Workshop on Formal Methods and Security Protocols (FMSP'99)*, vol. Trento (Italy), 1999.
- [22] H. Comon-Lundh and V. Cortier., "New decidability results for fragments of first-order logic and application to cryptographic protocols," *14th International Conference on Rewriting Techniques and Applications (RTA'2003). Valencia (Spain), Springer-Verlag.*, vol. 2706, pp. 148–164, 2003.
- [23] D. L. Mitchell, N. A. Durgin, P. D. Lincoln, J. C. Mitchell, and A. Scedrov, "Undecidability of bounded security protocols," 1999.
- [24] S. A. Schneider, H. Treharne, and N. Evans, "Chunks: Component verification in csp_{llb}," in *IFM*, pp. 89–108, 2005.
- [25] M. Abadi, "Secrecy by typing in security protocols," *Journal of the ACM*, vol. 46, pp. 611–638, 1998.
- [26] H. Comon-Lundh, V. Cortier, and E. Zalinescu, "Deciding security properties for cryptographic protocols. application to key cycles," *ACM Trans. Comput. Log.*, vol. 11, no. 2, 2010.
- [27] V. Cortier and S. Delaune, "Decidability and combination results for two notions of knowledge in security protocols," *J. Autom. Reasoning*, vol. 48, no. 4, pp. 441–487, 2012.
- [28] V. Cortier, S. Kremer, and B. Warinschi, "A survey of symbolic methods in computational analysis of cryptographic systems," *J. Autom. Reasoning*, vol. 46, no. 3-4, pp. 225–259, 2011.